



Inst. für Angew. Informationsverarbeitung

Prof. Dr. Franz Schweiggert
Michaela Weiss
Wolfgang Kaifler

23.11.2010
Lösung 3

Systemnahe Software I (WS 2010/2011)

Ausgabetermin: 30.11.2010

Aufgabe 1: Fragen (4 Punkte)

Überlegen Sie sich

Wie wird ein Mehrdimensionales Array (statisch allokiert) z.B. `a[4][3]` im Speicher abgelegt?

Lösung

Die Elemente werden nacheinander im Speicher abgelegt:

```
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| [0][0] | [0][1] | [0][2] | [1][0] | [1][1] | [1][2] | [2][0] | [2][1] | .....
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
```

Erklären Sie, was es mit dem *Zweierkomplement* auf sich hat!

Lösung

Das Zweierkomplement bietet eine Möglichkeit, negative Zahlen im Binärsystem darzustellen. Das Zweierkomplement benötigt wie das Einerkomplement keine Fallunterscheidung, ob mit negativen oder positiven Zahlen gerechnet wird. Positive Zahlen werden mit einer führenden 0 versehen und ansonsten nicht verändert.

Negative Zahlen werden mit einer führenden 1 dargestellt. Dies wird wie folgt erreicht: Alle Ziffern der positiven Zahl werden negiert, anschließend zum Ergebnis eine 1 hinzuaddiert.

$$-4 + 3 = -1$$

$$11111100 + 00000011 = 11111111$$

$$+4 - 4 = 0$$

$$00000100 + 11111100 = 1\ 00000000$$

Durch Abschneiden der 1 wird daraus 00000000

Aufgabe 2: Matrizen und Arrays (4 Punkte)

Lösen Sie auf Papier!

Was gibt folgendes Programm aus:

```
1 int v[2][3] = { {1,2,3},{4,5,6}};
2 char* cp = (char *) v;
3
4 int i;
5
6 printf("Durchlauf der Matrix via char-pointer:\n");
7 for(i = 0; i < 24 ; i++) {
8     printf("%u ", *cp);
9     cp++;
10 }
11 printf("\n");
```

Quellcode 1: fehlerhaft

Beachten Sie die Datentypen!

Lösung

Es wird ein zweidimensionales Array angelegt, welches mit 6 Integern zu je 4 Byte gefüllt wird. Die Elemente des Arrays liegen bekannterweise hintereinander im Speicher (6 Elemente zu je 4 Byte = 24 Byte)

Anschließend wird das Array auf Chars(= 1 Byte) gecastet, d.h. das ursprünglich aus Integern bestehende Array wird von nun an als Folge lauter Chars interpretiert.

Diese Folge kann nun in einer Schleife durchlaufen werden, wobei jedes Byte einzeln ausgegeben wird.

Da hier ursprünglich Integer gespeichert wurden, diese aber nun anders interpretiert werden, ist das ganze Architekturabhängig (vgl. Little- und Big-Endian). Auf einem Intel-System sollte das Programm folgendes ausgeben:

1 0 0 0 2 0 0 0 3 0 0 0 4 0 0 0 5 0 0 0 6 0 0 0

auf einem SPARC-System hingegen:

0 0 0 1 0 0 0 2 0 0 0 3 0 0 0 4 0 0 0 5 0 0 0 6

Viel Erfolg!